Complexity classes

defined by counting quantifiers*

Jacobo Torán
Dept. L.S.I.
Universidad Politecnica de Cataluña
Pau Gargallo 5
08028 Barcelona, Spain

Abstract:

We study the polynomial time counting hierarchy, a hierarchy of complexity classes related to the notion of *counting*. We investigate some of their structural properties, settling many open questions dealing with oracle characterizations, closure under boolean operations, and relations with other complexity classes. We develop a new combinatorial technique to obtain relativized separations for some of the studied classes, which imply absolute separations for some logarithmic time bounded complexity classes.

1. Introduction.

One of the main goals of complexity theory is the classification of computational problems in complexity classes according to the amount of resources these problems need. Probably the best known complexity classes are P and NP since both of them capture the complexity of many natural problems, and also because the long standing open question $P \stackrel{?}{=} NP$ has motivated most of the research in the area. The similarities in the definitions of the class NP and the recursion-theoretic class of the recursively enumerable sets (both can be characterized by an existential quantifier) provoked the "translation" of other recursion-theoretic notions into the field of complexity theory, and the analogous concept to the arithmetic hierarchy, the polynomial time hierarchy [St,77] was defined. The idea is a natural generalization of the class NP, and provided a good tool to classify more complex problems. It was taken also by many researchers as a frame for the study of structural complexity theory and the idea behind the hierarchy, the alternation of existential and universal quantifiers [Ch,Ko,St,81] influenced very much the work in the area.

Nevertheless there are many natural computational problems whose complexity cannot be modelized in terms of existential or universal quantifiers; on the other hand this

* This article is part of the Ph.D. Thesis of the author. Some of its results have been presented at the international conferences STRUCT'88 and ICALP'89.

complexity is captured by other complexity classes, more adapted to the idea of *counting*.

Following this motivation, Simon defines in [Sim,75] the class of threshold languages. A language L is in this class if there is a polynomial time Turing machine M such that for every input x, M has at least k accepting computation paths if and only if x is in L, where k is a fixed constant or fraction. This class is placed between NP and PSPACE and it is closely related to Valiant's class #P of functions that count the number of accepting paths in a nondeterministic Turing machine [Va,79]. It contains natural complete problems; a typical problem in this class is

$$\#\ \mathrm{SAT} = \{(F,k) \mid F \text{ is a boolean formula with at least } k \text{ satisfying assignments}\}$$

Simon also shows that the class of threshold languages is the same as the class PP, of languages accepted by polynomial time probabilistic Turing machines [Gi,77]. The languages in this class are those recognized by polynomial time bounded Turing machines which accept an input if and only if more than half of the computation paths accept.

In order to characterize the complexity of some languages called "games against nature", Papadimitriou [Pa,83] generalizes the idea of probabilistic machine and obtains the class PPSPACE of languages accepted by polynomial time bounded Turing machines which alternate between nondeterministic and probabilistic configurations. The class PPSPACE turns out to be equal to PSPACE. Papadimitriou shows that in the same way as a language L in PSPACE can be characterized by an alternating string of existential and universal quantifiers followed by a polynomial time predicate, they can also be formulated by alternating the existential quantifier and R, a probabilistic (or random) quantifier:

$$x \in L \Longleftrightarrow \exists^p x_1 R^p x_2 \exists^p x_x \ldots P(x_1, x_2, \ldots, x_n)$$

where $R^{p(n)} x_1 P(x_1)$ means that there exist more than half of the strings of length $p(n)$ satisfying the predicate P.

One step further is taken by Wagner [Wa,86] when he defines the counting hierarchy (CH) in a similar way as the polynomial time hierarchy (PH), trying to classify the complexity of certain combinatorial problems in which counting is involved. Instead of using the probabilistic quantifier R, Wagner introduces the quantifier $\mathbf{C}$ inspired in the idea of threshold machines. As we will see more formally, $\mathbf{C}_{f(x)}^{p(n)}\ y\ P(y)$ means that there are at least $f(x)$ strings y of length $p(n)$ satisfying predicate P. This quantifier is equivalent to the probabilistic one, R, in the same way that the probabilistic machines recognize the same languages as the threshold machines. The hierarchy arises in a natural way combining the counting quantifier not only with the existential quantifier, as in [Pa,83], but also with the universal one. The counting hierarchy turns out to be a very useful tool to express the complexity of many natural problems. It contains the polynomial time hierarchy and is included in PSPACE. Wagner shows that every level of CH has complete problems and proves some other results about the hierarchy.

As we have already mentioned, many concepts in complexity theory are direct "translations" of the same concepts in the recursive function theory to the polynomial time case (ideas like reduction, polynomial time hierarchy, oracles, etc. are taken from the same ideas in the theory of recursive functions). It is interesting to observe that the polynomial counting quantifier is particular to complexity theory, since the analogous concept in re-

cursive function theory, the unbounded counting quantifier, is equivalent to an unbounded existential quantifier. In our opinion, the lack of a parallel concept in recursion theory has determined the late appearence of the concept in complexity theory.

As we have said, the counting hierarchy has great importance for the classification of a variety of computation problems. Nevertheless its structural properties have never been studied in depth, and it was assumed to behave in a similar way as other better known hierarchies, like the polynomial time hierarchy. This has been shown to be true only to a certain extent. In this work we try to complete this knowledge, investigating different aspects of CH, and solving some open problems related to the hierarchy.

The article is divided into different sections. After introducing notation and preliminaries in sections 3 and 4 we basically continue the work started by Wagner when he defined the polynomial counting hierarchy. We study the boolean properties of the classes in CH, showing that a class is closed under union and intersection if the first quantifier defining it is either $\exists$ or $\forall$, and closed under complement and symmetric difference if it is the $\mathbf{C}$ quantifier. Classes whose characterization starts by a $\mathbf{C}$ quantifier do not seem to be closed under union and intersection, and this fact makes that the classes in CH behave in a very different way than the classes in PH. We also study the "unbounded cartesian product" operation which can be considered as a certain kind of unbounded intersection, showing that the classes closed under unbounded cartesian product coincide with the classes closed under intersection, and also that the closure under this operation of a class whose characterization starts with quantifier $\mathbf{C}$ implies certain collapse result in CH. Using these results we are able to characterize the counting hierarchy in terms of nondeterministic and probabilistic machines with access to oracles. This characterization was only known for the classes of type Σ_k^p, (oracle characterization of the PH) and $\mathbf{C}\Sigma_k^p$, [Wa,86]. Our result completes the characterization for every class.

Section 5 is motivated by the problem of whether the classes studied can be separated. We introduce a new combinatiorial method to obtain relativized separations of the counting classes defined in the previous sections. Although counting classes have been separated from the polynomial time hierarchy before, [An,80], [Ya,85], [Hå,86], to our knowledge this is the first time that counting classes have been separated from other counting classes. The technique used to obtain our results is new since the methods from previous relativizations do not seem to work for counting complexity classes. The idea is to diagonalize gathering the number of accepting computation paths of the oracle Turing machines in combinatorial formulas in which the oracle is a variable, and then argue over the formulas using combinatorial techniques and the fact that our machines are polynomial time bounded. We present three relativizations separating NP from $\mathbf{G}=$ (exact counting), NP from $\oplus$P and $\oplus$P from PP. As a consequence we obtain relativizations in which the three classes NP, $\oplus$P, and $\mathbf{G}=$ are incomparable, $\oplus$P and PP are incomparable, and NP and $\mathbf{G}=$ are strictly contained in PP. These separations also imply a relativization in which PP is different from PSPACE, solving an open problem proposed by Angluin in [An,80], as well as relativized separations of the lower levels of the counting hierarchy. Another consequence of the relativizations presented is the absolute separation of log-time complexity classes.

We include at the end of the article a section of conclusions and further research areas.

2. Notation and preliminaries

The notation used in this article is the standard one in stuctural complexity theory, and when new concepts are used, a definition of them is included. However trying to avoid any possible confusion, we include a short summary of notation.

The sets that we will considered are languages over some fixed alphabet Σ. For a set $A \subseteq \Sigma^\star$, $||A||$ will represent the cardinality of A; and for a string $x \in \Sigma^\star$, $|x|$ will denote its length. The complement of a set A will be denoted $\overline{A}$, and its characteristic function χ_A. Easily computable pairing functions are assumed, and denoted by angular parenthesis, as in $\langle x, y \rangle$. The marked union or join of two sets A and B is defined as $A \oplus B = \{0x \mid x \in A\} \cup \{1x \mid x \in B\}$.

Our model of computation will be the multi-tape Turing machine. If M is a Turing machine, $\mathrm{L}(M)$ represents the language accepted by M. If M is a nondeterministic Turing machine, for a pair of strings x, y, $M_y(x)$ represents the computation of M on input x following computation path y; if M has access to some oracle, $M_y^A(x)$ represents the computation on x following path y and oracle A and $L(M, A)$ represents the language accepted by machine M with oracle A. In some parts of this work it is assumed without explicit mention that the input tape alphabet of Turing machines is $\{0, 1\}$.

For a definition of the language classes P, NP, PSPACE, UP, LOGSPACE and the ones in the polynomial time hierarchy we refer the reader to the books written on the subject [Ba,Di,Ga,88], [Sc,85], [Wa,We,86]. PP [Gi,77] is the class of language recognized by polynomial time nondeterministic machines that accept an input x if and only if more than one half of all the computation paths of the machine accept x. For a language class K, P^K represents the class of languages computed by a polynomial time machine with access to an oracle in K. Moreover, $\mathrm{P}^{K[(\log n)]}$ denotes the same class of languages, with the restriction that the polynomial time machine can only make $O(\log n)$ queries to the oracle.

We will also consider different function classes. We will denote by FP the class of polynomial time computable functions and by FP^A the class of functions computable in polynomial time by a deterministic machine with access to oracle A.

The polynomial time reducibilities used are the many-one reducibility ($\leq_m^p$), and the Turing reducibility ($\leq_T^p$), whose definition can be seen in the mentioned books.

3. The counting hierarchy

As mentioned before, the polynomial time counting hierarchy was first introduced by Wagner [Wa,86] as a tool to classify certain combinatorial problems in which counting is involved. However, many problems were left open by his work; among them, many of the structural properties of this hierarchy, such as equivalent definitions by oracle machines (similar to the case of the polynomial time hierarchy), closure under boolean operations, and ability to extract information from oracle sets. In this and the following section we

address these problems, extending the work of Wagner.

We first study closure properties of the classes in the hierarchy. We show that they depend only on the first quantifier of the string of alternating quantifiers defining the class, being a class closed under complement and symmetric difference if its first quantifier is a counting one, and closed under union and intersection if the first quantifier is either existential or universal. We also study the closure of the classes under "unbounded cartesian product", an operation which is needed to obtain other characterizations of CH. We show that the classes that are closed under unbounded cartesian product, coincide with the classes closed under intersection. Using these results, we also prove that for certain classes K in CH, a deterministic oracle machine asking just one question to a set in K, can only recognize sets in K.

The following definitions are taken from the article [Wa,86].

Definition 3.1: The polynomial counting quantifier $\mathbf{C}$, is defined in the following way; for a function $f : \Sigma^\star \longrightarrow \mathbb{N}, f \in$FP, a polynomial p and a two argument predicate Q,

$$\mathbf{C}^p_{f(x)}\ y\ :\ Q(x,y) \Longleftrightarrow ||\{y :\ |y| \leq p(|x|) \text{ and } Q(x,y)\}|| \geq f(x).$$

If K is a language class, for any set A, $A \in \mathbf{C}K$ if there is a function f in FP, s.t. for every x, $f(x) > 0$, a polynomial p and a language $B \in K$ such that for any $x \in \Sigma^\star$

$$x \in A \Longleftrightarrow \mathbf{C}^p_{f(x)}\ y\ : \langle x, y\rangle \in B$$

Recall that bounded quantifiers have been used before to define complexity classes. For example, in [Wr,77] a characterization of the polynomial time hierarchy in terms of the polynomially bounded existencial and universal quantifiers $\exists^p$ and $\forall^p$ is given.

We alternate now the polynomial counting quantifier $\mathbf{C}$ with the existential and the universal quantifiers in order to define the counting hierarchy.

Definition 3.2: The polynomial counting hierarchy (CH) is the smallest family of language classes satisfying:

i/ P$\in$CH

ii/ If $K \in$CH then $\exists^p K$, $\forall^p K$ and $\mathbf{C}^p K \in$CH.

Since in this section we will talk only about quantifiers ranging over strings of polynomial length, we drop the superscript p from all the quantifiers. Also, for simplicity, $\mathbf{C}$ will denote the class $\mathbf{C}$P, and the context will make clear when we talk about a quantifier and when about a language class.

The next lemma shows that the threshold function of the $\mathbf{C}$ quantifier can be changed to strictly more than one half of the possible quantified strings. As a consequence, the

class $\mathbf{C}$ is the same as the class PP of languages accepted by probabilistic Turing machines. This fact has been observed in [Sim,75], [Wa,86].

Lemma 3.3:
i/ $\mathbf{C} = \mathrm{PP}$
ii/ For any class K in CH and any function $f : \mathbb{N} \longrightarrow \mathbb{N}, f \in \mathrm{FP}$ $\mathbf{C}^p_f K \subseteq \mathbf{C}^p_{(2^{p(n)-1}+1)} K$.

To locate CH with respect to other complexity classes, we observe the following properties:

Lemma 3.4: [Wa,86]
a/ Every language in CH can be accepted in polynomial space.
b/ PH$\subseteq$CH
c/ For every class K in CH, $\exists K \cup \forall K \subseteq \mathbf{C}K \subseteq \exists \mathbf{C}K \cap \forall \mathbf{C}K$.
d/ Every class in CH is closed under $\leq^p_m$ reducibility.
e/ Every class in CH has $\leq^{log}_m$ complete problems.

Some easy closure properties of the classes in CH that will be used later are:

Lemma 3.5: For any class K in CH, any sets L_1, L_2 in K and any set P_1 in P:
i/ $L_1 \oplus L_2 \in K$.
ii/ $L_1 \times \mathbb{N} \in K$.
iii/ $L_1 \cap P_1 \in K$.

Next we define the "exact" counting quantifier $\mathbf{G}$, which is a little different than $\mathbf{C}$ in its definition, but as we will see, has very different properties. This difference between $\mathbf{C}$ and $\mathbf{G}$, will be crucial in the proof of our results. $\mathbf{G}$ was also defined for the first time in [Wa,86]. In section 5 we present a relativization in which $\mathbf{G} \neq \mathbf{C}$ (considered as language classes).

Definition 3.6: For a function $f : \mathbb{N} \longrightarrow \mathbb{N}, f \in \mathrm{FP}$, a polynomial p and a two argument predicate Q,

$$\mathbf{G}^p_{f(x)}\ y\ :\ Q(x,y) \Longleftrightarrow ||\{y :\ |y| \leq p(|x|) \text{ and } Q(x,y)\}|| = f(x).$$

The definition of $\mathbf{G}K$ for a language class K is analogous to the definition of $\mathbf{C}K$, using the new quantifier. Also lemma 3.5 holds for any class K in CH, and any sets L_1, L_2 in $\mathbf{G}K$.

In order to show the closure under certain boolean operations of the classes in CH, we need to prove the following results, which will be improved later in corollary 3.11.

Lemma 3.7: For any class K in CH,

i/ $\exists \mathbf{C}K \subseteq \exists \mathbf{G}\!\!\!-K$.

ii/ $\mathbf{C}\mathbf{C}K \subseteq \mathbf{C}\mathbf{G}\!\!\!-K$.

iii/ $\mathbf{G}\!\!\!-\mathbf{C}K \subseteq \mathbf{G}\!\!\!-\mathbf{G}\!\!\!-K$.

iv/ $\mathbf{G}\!\!\!-K \subseteq \mathbf{C}K \triangle \mathbf{C}K$.

Proof: i/: Let K be a class in CH and L a set in $\exists \mathbf{C}K$. There is a function f in FP and a set $A \in K$ such that for every $x \in \Sigma^\star$

$$\begin{aligned} x \in L &\iff \exists y \mathbf{C}_{f(x,y)}\ z\ :\ \langle x,y,z\rangle \in A \\ &\iff \exists y \exists v \mathbf{G}\!\!\!-_{f(x,y)}\ z\ :\ [\langle x,y,z\rangle \in A \text{ and } v \leq z] \end{aligned}$$

ii/ and iii/: Analogous to i/.

iv/: Let K be a class in CH and L a set in $\mathbf{G}\!\!\!-K$. There is a function f in FP and a set $A \in K$ such that for every $x \in \Sigma^\star$

$$x \in L \iff \mathbf{G}\!\!\!-_{f(x)}\ y\ :\ \langle x,y\rangle \in A$$

Let $L_1 = \{x : \mathbf{C}_{f(x)}\ y\ :\ \langle x,y\rangle \in A\}$ and $L_2 = \{x : \mathbf{C}_{f(x)+1}\ y\ :\ \langle x,y\rangle \in A\}$.

L_1 and L_2 are in $\mathbf{C}K$ and $L = L_1 \triangle L_2$. □

We will improve the above result in corollary 3.11 after proving certain boolean properties of the classes in CH.

In [Wa,86] it is mentioned that the class $\mathbf{G}\!\!\!-$ is closed under intersection. This can be extended to any class starting by the $\mathbf{G}\!\!\!-$ quantifier and to a certain kind of unbounded intersection, the unbounded cartesian product. This extension will be necessary for the oracle characterization of CH in the next section.

Definition 3.8: For any set L define the unbounded cartesian product of L, $L^\times$, as the set

$$L^\times = \{\langle x_1, x_2, \ldots, x_k\rangle : \bigwedge_i x_i \in L\}$$

Theorem 3.9: For any class K in CH, $\mathbf{G}\!\!\!-K$ is closed under unbounded cartesian product.

Proof: Let L be a set in $\mathbf{G}\!\!\!-K$. There is a set A in K, a function f in FP , and a polynomial p such that for any $x \in \Sigma^\star$ $f(x) < 2^{p(|x|)}$ and

$$x \in L \iff \mathbf{G}\!\!\!-_{f(x)}\ y, |y| \leq p(|x|)\ :\ \langle x,y\rangle \in A$$

Given a sequence of strings $\langle x_1, x_2, \ldots, x_k\rangle$ let $m = \max\{|x_1|, \ldots |x_k|\}$, and define
$g(x_1, \ldots x_k) = f(x_1) + f(x_2)2^{p(m)+1} + f(x_3)2^{2p(m)+2} + \ldots + f(x_k)2^{(k-1)(p(m)+1)}$.

Notice that from $g(x_1, \ldots x_k)$ it is possible to recover the unique values of $f(x_1) \ldots f(x_k)$.

$$\begin{aligned}
&\langle x_1, x_2, \ldots, x_k \rangle \in L^{\times} \Longleftrightarrow \\
\Longleftrightarrow &\mathbf{G}_{f(x_1)}\ y_1, |y_1| \leq p(|x_1|), \langle x, y_1 \rangle \in A \wedge \ldots \wedge \mathbf{G}_{f(x_k)}\ y_k, |y_k| \leq p(|x_k|), \langle x_k, y_k \rangle \in A \\
\Longleftrightarrow &\mathbf{G}_{g(x_1, x_2 \ldots x_k)}\ \langle z_1, z_2, z_3 \rangle \text{ (there exists an } i \text{ such that } z_1 = x_i \text{ and } |z_2| \leq p(|x_i|) \\
&\text{and } \langle x_i, z_2 \rangle \in A \text{ and } |z_3| = (i-1)(p(m)+1).
\end{aligned}$$

This is true because we have multiplied the witnesses of $x_i \in L$ in such a way that there must be exactly $f(x_i)2^{(i-1)(p(m)+1)}$ of them for every x_i. It follows

$$\begin{aligned}
&\langle x_1, x_2, \ldots, x_k \rangle \in L^{\times} \Longleftrightarrow \\
&\mathbf{G}_{g(x_1, x_2, \ldots, x_k)}\ \langle z_1, z_2, z_3 \rangle\ :\ \langle z_1, z_2, z_3 \rangle \in A'
\end{aligned}$$

being A' in K, which implies $L^{\times} \in \mathbf{G}K$. □

We prove now the main result of this section.

Theorem 3.10: Let K be the class in CH characterized by the quantifiers $Q_1 Q_2 \ldots Q_k$

i/ If Q_1 is either $\exists$ or $\forall$ then the class K is closed under intersection and union.
ii/ If Q_1 is **C** then the class K is closed under complement and symmetric difference.
iii/ If Q_1 is $\exists$ ($\forall$) then co-$K \subseteq \forall K$ (co-$K \subseteq \exists K$).
(Fact iii/ is a technical property needed for the proof of the rest of the theorem.)

Proof: The proof is by induction over k, the minimum number of quantifiers characterizing the class K. It is divided in different cases, in order to cover all types of quantifiers.
Induction basis: $k = 1$
i/ If Q_1 is either $\exists$ or $\forall$ then trivially K is closed under intersection and union.
ii/ If Q_1 is **C** then
K is closed under complement [Gi,77], and
K is closed under symmetric difference [Ru,85].
iii/ If Q_1 is $\exists$ then co-$K = \Pi_1$ and trivially $\Pi_1 \subseteq \Pi_2$. If Q_1 is $\forall$ then co-$K = \Sigma_1$ and trivially $\Sigma_1 \subseteq \Sigma_2$.

Induction step: $k \Rightarrow k+1$
Let K be in CH, $K = Q_1 K' = Q_1 Q_2 K''$, (if Q_1 is either $\exists$ or $\forall$ then $Q_1 \neq Q_2$.)
i/ Q_1 is $\exists$
Intersection.
a/ Q_2 is $\forall$
K is the class $\exists\forall K''$ with K'' in CH and characterized by $k-1$ quantifiers. Let L_1 and L_2 be two sets in K. There are two sets B_1, B_2 in $\forall K''$, and a polynomial p such that for $i = 1, 2$ and for any $x \in \Sigma^{\star}$

$$x \in L_i \Longleftrightarrow \exists y\ |y| = p(|x|) \text{ and } \langle x, y \rangle \in B_i$$

$$x \in L_1 \cap L_2 \iff \exists \langle y_1, y_2 \rangle \ : \ (\langle x, y_1 \rangle, \langle x, y_2 \rangle) \in (B_1 \times \mathbb{N}) \cap (\mathbb{N} \times B_2) \qquad (1)$$

By lemma 3.5, $(B_1 \times \mathbb{N}), (\mathbb{N} \times B_2) \in \forall K''$; by induction hypothesis $(B_1 \times \mathbb{N}) \cap (\mathbb{N} \times B_2) \in \forall K''$, and $L_1 \cap L_2 \in \exists \forall K''$. A similar argument works for $K = \forall \exists K''$.

b/ Q_2 is $\mathbf{C}$

There are two sets B_1, B_2 in $\mathbf{C}K''$ satisfying the above formula (1). By lemma 3.7 these two sets can be substituted by two new ones D_1, D_2 in $\mathbf{G}_{\neq}K''$, and combining (the $\mathbf{G}_{\neq}$ version of) lemma 3.5 and lemma 3.9, $(D_1 \times \mathbb{N}) \cap (\mathbb{N} \times D_2) \in \mathbf{G}_{\neq}K''$. By the third part of lemma 3.7 $(D_1 \times \mathbb{N}) \cap (\mathbb{N} \times D_2)$ is in the symmetric difference of two sets in $\mathbf{C}K''$ which by induction hypothesis is in $\mathbf{C}K''$. It follows that $L_1 \cap L_2 \in \exists \mathbf{C}K''$.

Union.

a/ Q_2 is $\forall$

The proof is completely analogous to the intersection case.

b/ Q_2 is $\mathbf{C}$

Let B_1, B_2 be the sets in $\mathbf{C}K''$ defined in the intersection case. For every $x \in \Sigma^\star$

$$x \in L_1 \cup L_2 \iff \exists \langle y_1, y_2, a \rangle : a \in \{0,1\} \text{ and } a(\langle x, y_1 \rangle, \langle x, y_2 \rangle) \in (B_1 \times \mathbb{N}) \oplus (\mathbb{N} \times B_2)$$

by lemma 3.5, $(B_1 \times \mathbb{N}) \oplus (\mathbb{N} \times B_2) \in \mathbf{C}K''$ and $L_1 \cup L_2 \in \exists \mathbf{C}K''$.

If Q_1 is $\forall$ then the proof follows from the above $\exists$-case using the complementary classes; for example for the intersection, if we have two sets L_1 and L_2 in $\forall K'$, then $\overline{L_1}$ and $\overline{L_2}$ are in $\exists$co-K' and $\overline{L_1} \cup \overline{L_2} \in \exists$co-$K'$. $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}} \in \forall K'$. The union case is analogous.

ii/ Q_1 is $\mathbf{C}$

Complement.

Let L be a set in $\mathbf{C}K$. There is a set A in K', a function f in FP and a polynomial p, such that for any $x \in \Sigma^\star$

$$\begin{aligned} x \in \overline{L} &\iff \neg(\mathbf{C}_{f(x)} \ y \ |y| = p(|x|) \ : \ \langle x, y \rangle \in A \qquad (2) \\ &\iff \mathbf{C}_{2^{p(|x|)} - f(x)} \ y \ |y| = p(|x|) \ : \ \langle x, y \rangle \notin A \end{aligned}$$

a/ Q_2 is $\forall$

The set A in the above formula is in $\forall K''$. There is a set B in K'' such that

$$\begin{aligned} x \in \overline{L} &\iff \mathbf{C}_{2^{p(|x|)} - f(x)} \ y \ |y| = p(|x|) \ \exists z \ : \ \langle x, y, z \rangle \notin B \\ &\iff \mathbf{C}_{2^{p(|x|)} - f(x)} \ \langle y, z \rangle \ : \ [\langle x, y, z \rangle \notin B \text{ and } \forall z' (z' < z \Rightarrow \langle x, y, z' \rangle \in B)] \\ &\iff \mathbf{C}_{2^{p(|x|)} - f(x)} \ \langle y, z \rangle \ : \ \langle x, y, z \rangle \in (\overline{B} \cap D) \end{aligned}$$

being D a set in $\forall K''$. $\overline{B} \in \forall K''$ (by fact iii/ if $K'' = \exists K'''$, or by fact ii/ if $K'' = \mathbf{C}K'''$). By induction hypothesis $\overline{B} \cap D \in \forall K''$, and then $\overline{L} \in \mathbf{C} \forall K''$. We have shown co-$\mathbf{C} \forall K'' \subseteq \mathbf{C} \forall K''$ It follows co-$\mathbf{C} \forall K'' = \mathbf{C} \forall K''$.

b/ Q_2 is $\exists$

co-$\mathbf{C}\exists = \mathbf{C}\exists$ since co-$\mathbf{C}\exists K'' = \mathbf{C}$co-$\exists K'' = \mathbf{C}\forall$co-$K'' =$co-$(\mathbf{C}\forall$co-$K'') = \mathbf{C}\exists K''$.

c/ Q_2 is $\mathbf{C}$

In this case, set A in (2) is in the class $\mathbf{C}K''$ which by hypothesis is closed under complement.

Symmetric difference.

Let L_1, L_2 be two sets in $\mathbf{C}K'$. By lemma 3.3 we can change the threshold of the first quantifier to be more than half of the possible strings. There are two sets B_1, B_2 in K' and a polynomial p such that for any $x \in \Sigma^\star$

$$x \in L_i \iff \mathbf{C}_{2^{p(|x|)-1}+1}\ y\ |y| = p(|x|) \quad : \langle x, y\rangle \in B_i$$

Following the same idea as in Russo's proof that PP is closed under symmetric difference [Ru,85], let $x \in \Sigma^\star$, and let a_1 and a_2 be the two integers (not necesarily positive) such that

$$||\{y_i \ : \ \langle x, y_i\rangle \in B_i\}|| = 2^{p(n)-1} + a_i$$

Let

$$t = ||\{\langle y_1, y_2\rangle \ : \ [(\langle x, y_1\rangle \in B_1 \text{ and } \langle x, y_2\rangle \notin B_2) \text{ or } (\langle x, y_1\rangle \notin B_1 \text{ and } \langle x, y_2\rangle \in B_2)]||$$

$$t = (2^{p(n)-1} + a_1)(2^{p(n)-1} - a_2) + (2^{p(n)-1} - a_1)(2^{p(n)-1} + a_2) = 2^{p(n)-1} - 2a_1a_2$$

If $x \in L_1 \triangle L_2$ then either ($a_1 \geq 1$ and $a_2 < 1$) or ($a_1 < 1$ and $a_2 \geq 1$). In both cases $t \geq 2^{2p(n)-1}$.

If $x \notin L_1 \triangle L_2$ then either ($a_1 \geq 1$ and $a_2 \geq 1$) or ($a_1 < 1$ and $a_2 < 1$), and in both cases $t < 2^{2p(n)-1}$. Therefore

$$\begin{aligned} x \in L_1 \triangle L_2 &\iff \mathbf{C}_{2^{2p(|x|)-1}+1}\ \langle y_1, y_2\rangle \ : \ [(\langle x, y_1\rangle \in B_1 \text{ and } \langle x, y_2\rangle \notin B_2) \\ &\qquad \text{or } (\langle x, y_1\rangle \notin B_1 \text{ and } \langle x, y_2\rangle \in B_2)] \\ &\iff \mathbf{C}_{2^{2p(|x|)-1}+1}\ \langle y_1, y_2\rangle \ : \ (\langle x, y_1\rangle, \langle x, y_2\rangle) \in (B_1 \times \mathbb{N}) \triangle (\mathbb{N} \times B_2) \end{aligned} \quad (3)$$

a/ Q_2 is $\exists$

The above sets B_i are then in the class $\exists K''$. There are two sets D_1, D_2 in K'' such that

$$\begin{aligned} x \in L_1 \triangle L_2 \iff \mathbf{C}_{2^{2p(|x|)-1}+1}\ \langle y_1, y_2\rangle \ : \ [\exists z \ : \ &(\langle x, y_1, z\rangle \in D_1 \text{ or } \langle x, y_2, z\rangle \in D_2) \\ &\text{and } (\forall z_1, z_2 \ : \ \langle x, y_1, z_1\rangle \notin D_1 \text{ or } \langle x, y_2, z_2\rangle \notin D_2)] \end{aligned}$$

$$\begin{aligned} \iff \mathbf{C}_{2^{2p(|x|)-1}+1}\ \langle y_1, y_2, z\rangle \ : \ [&(\langle x, y_1, z\rangle \in D_1 \text{ or } \langle x, y_2, z\rangle \in D_2) \\ &\text{and } (\forall z_1, z_2 \ : \ \langle x, y_1, z_1\rangle \notin D_1 \text{ or } \langle x, y_2, z_2\rangle \notin D_2) \\ &\text{and } (\forall z' \ : z' < z \Rightarrow (\langle x, y_1, z'\rangle \notin D_1 \text{ and } \langle x, y_2, z'\rangle \notin D_2))] \end{aligned}$$

It follows that $L_1 \triangle L_2 \in \mathbf{C}\forall$co-$K''$. As we have seen this class is closed under complements and $L_1 \triangle L_2 \in$co-$(\mathbf{C}\forall$co-$K'') = \mathbf{C}\exists K''$.

b/ Q_2 is $\mathbf{C}$

By lemma 3.5,the sets $(B_1 \times \mathbb{N})$ and $(\mathbb{N} \times B_2)$ in (3) are in $\mathbf{C}K''$ and by induction hypothesis the symmetric difference of these sets is in $\mathbf{C}K''$ and $L_1 \triangle L_2 \in \mathbf{CC}K''$.

iii/ Q_1 is $\exists$ ($\forall$)

a/ Q_2 is $\forall$ ($\exists$)

K is the class $\exists K'$ and $K' = \forall K''$, by induction hypothesis co-$K' \subseteq \exists K'$ and therefore co-$K = \forall$co-$K' \subseteq \forall\exists K' = \forall K$.

b/ Q_2 is $\mathbf{C}$

By induction hypothesis $K' =$co-K'. Therefore co-$K' \subseteq K' \subseteq \forall K'$.

□

Later we will need a stronger version of part ii/ of the theorem; for any class K in CH, $\exists K$ and $\forall K$ are closed under unbounded cartesian product. The proof of this fact is analogous to the above one.

Corollary 3.11: For any class K in CH

i/ $\exists \mathbf{C}K = \exists \mathbf{G}\text{-}K$.

ii/ $\mathbf{CC}K = \mathbf{C}\mathbf{G}\text{-}K$.

iii/ $\mathbf{G}\text{-}\mathbf{C}K = \mathbf{G}\text{-}\mathbf{G}\text{-}K$.

iv/ $\mathbf{G}\text{-}K \subseteq \mathbf{C}K$.

The next result shows that it is not likely that the classes starting with quantifier $\mathbf{C}$ are closed under unbounded cartesian product. As mentioned before, for a language class K, $\mathrm{P}^{K[\mathrm{O}(\log n)]}$ denotes the class of languages accepted by a polynomial time deterministic machine that queries an oracle in K at most a logarithmic number of times.

Theorem 3.12: For any class K in CH, if $\mathbf{C}K$ is closed under unbounded cartesian product, then $\mathbf{C}K = \mathrm{P}^{\mathbf{C}K[\mathrm{O}(\log n)]}$.

Proof: The inclusion from left to right is straightforward, for the other one, let K be a class in CH, and $L \in \mathrm{P}^{\mathbf{C}K[\mathrm{O}(\log n)]}$ via a polynomial time deterministic Turing machine M quering at the most $c \log n$ times an oracle $A \in \mathbf{C}K$. For a given input x, we can encode the oracle answers from A on the computation of M, by a string of $c\log(|x|)$ bits y.

$$x \in L \Longleftrightarrow \exists y, |y| \leq c\log(|x|) \\ (M^y(x) \wedge f_1(x,y) \in A \oplus \overline{A} \wedge f_2(x,y) \in A \oplus \overline{A} \ldots f_{|y|}(x,y) \in A \oplus \overline{A})$$

where $M^y(x)$ means that M accepts x, following the oracle answers encoded in y, and $f_i(x,y) = \langle w, a\rangle$ being w the i-th query that M on input x and following y makes to the oracle, and a the i-th bit of y. We can write the above expression as

$$x \in L \Longleftrightarrow \exists y, |y| \leq c\log(|x|)(M^y(x) \wedge \langle f_1(x,y), f_2(x,y), \ldots, f_{|y|}(x,y)\rangle \in (A \oplus \overline{A})^{\times})$$

denote $\langle f_1(x,y), f_2(x,y), \ldots, f_{|y|}(x,y)\rangle$ by $h(x,y)$, and $(A \oplus \overline{A})^{\times}$ by B. By the closure of $\mathbf{C}K$ under complements, unbounded cartesian product (hypothesis) and intersection with polynomial time predicates, the set B belongs to $\mathbf{C}K$, and

$$x \in L \Longleftrightarrow \exists y, |y| \leq c\log(|x|),\ M^y(x)\ \wedge\ h(x,y) \in B$$

Since the y in the quantifier has logarithmic length, we can avoid it by writing explicitely all the strings of this length.

$$x \notin L \Longleftrightarrow \langle h(x,y_0), h(x,y_1), \ldots, h(x,y_{|x|^c})\rangle \notin (\overline{B})^{\times}$$

Being y_i the i-th string of length $\leq c\log(|x|)$ in lexicographical order. Using again that $\mathbf{C}K$ is closed under complements and the hypothesis, it follows that $L \in \mathbf{C}K$ and $\mathrm{P}^{\mathbf{C}K[\mathrm{O}(\log n)]} \subseteq \mathbf{C}K$. □

Another consequence of theorem 3.10 is that deterministic polynomial time oracle Turing machines that can make just one question to an oracle in a CH class whose characterization starts with the counting quantifier, can only recognize those languages in the class.

Corollary 3.13: For any class K in CH, $\mathrm{P}^{\mathbf{C}K[1]} = \mathbf{C}K$.

Proof: The inclusion from left to right is straightforward; for the converse, let A be a m-complete set in $\mathbf{C}K$, [Wa,86]. It is clear that for any language L in $\mathrm{P}^{\mathbf{C}K[1]}$, $L \leq_m^p A \oplus \overline{A}$. By the closure of $\mathbf{C}K$ under complements and m-reducibility, $L \in \mathbf{C}K$.

□

4. Characterizing the counting hierarchy with oracles

In this section, we show that the counting hierarchy coincides with the hierarchy obtained by iterating nondeterministic and probabilistic machines with oracles; We unify both concepts by giving an oracle characterization of the hierarchy, similar to the oracle characterization of the polynomial time hierarchy; the difference is that here instead of using only nondeterministic Turing machines, we also use probabilistic machines. This characterization extends a result from [Wa,86] where it is shown that for any class Σ_k^p in PH, $\mathrm{PP}^{\Sigma_k^p} = \mathbf{C}\Sigma_k^p$.

Theorem 4.1: For any class K in CH,
i/ $\mathrm{PP}^K = \mathbf{C}K$.
ii/ $\mathrm{NP}^{\exists K} = \mathrm{NP}^{\forall K} = \exists\forall K$ and $\mathrm{NP}^{\mathbf{C}K} = \exists\mathbf{C}K$.

Statement ii/ is divided in two cases depending of the quantifier characterization of the class in the oracle.

Proof: We prove i/, we will see later that the proof of ii/ is completely analogous.
$\supseteq$ **:** Straightforward.

$\subseteq$ **:** Let K be a class in CH, $K = Q_1 K'$, being Q_1 the first quantifier characterizing the class, and K' in CH.

Let L be a set in PP^K. There is a probabilistic Turing machine M, a polynomial p bounding the computation time of M, and a set A in K such that $L = L(M, A)$. For every $x \in \Sigma^\star$,

$$x \in L \Longleftrightarrow \mathbf{C}_{2^{p(|x|)-1}+1}\ y\ :\ M_y^A(x) \text{ accepts}$$

a/ Q_1 is $\exists$
Let B_1 be the set
$B_1 = \{\ \langle x, y, (q_1, z_1) \ldots (q_k, z_k), q_{k+1}, \ldots, q_m \rangle\ :$
$(\star)$ $m < p(|x|)$ and M with input x, following computation path y, asks all questions q_i in the list (not neccessarily in the same order), and answering them "yes" if $i \leq k$ and "no" if $i > k$, M accepts, and
$(\star\star)$ for $i = 1 \ldots k,\ q_i \in A$ and z_i is the smallest string witnessing this fact, and
$(\star\star\star)$ for $i = k+1 \ldots m,\ q_i \in \overline{A}$ $\}$

We claim that $B_1 \in \forall\text{co-}K'$. Since for every x and for every y such that $M_y^A(x)$ accepts, there is exactly one string v such that $\langle x, y, v \rangle \in B_1$, it is clear that

$$x \in L \Longleftrightarrow \mathbf{C}_{2^{p(|x|)-1}+1}\ w\ :\ \langle x, w \rangle \in B_1$$

It is only left to show that $B_1 \in \forall\text{co-}K'$. $(\star)$ can be checked in polynomial time. Since $\overline{A}$ is in $\forall\text{co-}K'$ and this class is closed under unbounded cartesian product, $(\star\star\star)$ is a predicate in $\forall\text{co-}K'$. Condition $(\star\star)$ can be written

$$\text{for } i = 1 \ldots k\ [\langle q_i, z_i \rangle \in D \text{ and } \forall z\ :\ (z < z_i \Rightarrow \langle q_i, z \rangle \notin D)]$$

being D a set in K'. By theorem 3.10, the predicate between [ ] is in $\forall\text{co-}K'$. Condition $(\star\star)$ is therefore an unbounded cartesian product of predicates in $\forall\text{co-}K'$, and by theorem 3.10 it is a predicate in $\forall\text{co-}K'$ and $B_1 \in \forall\text{co-}K'$.

We have shown $\mathrm{PP}^{\exists K'} \subseteq \mathbf{C}\forall\text{co-}K'$, but by theorem 3.10, $\mathbf{C}\forall\text{co-}K' = \mathbf{C}\exists K' = \mathbf{C}K$.

b/ Q_1 is $\mathbf{C}$.
The set A is in the class $\mathbf{C}K'$. By theorem 3.10 and lemma 3.5, the set $A \oplus \overline{A}$ is also in $\mathbf{C}K'$, and there is a function $f \in \mathrm{PF}$ and a set $D \in K'$ such that for every $u \in \Sigma^\star$,

$$u \in A \oplus \overline{A} \Longleftrightarrow \mathbf{C}_{f(u)}\ v\ :\ \langle u, v \rangle \in D$$

Let B_2 be the set
$B_2 = \{\ \langle x, y, (q_1, a_1, z_1) \ldots (q_m, a_m, z_m) \rangle\ :$

($\star$) $m < p(|x|)$ and M with input x, following computation path y, asks all questions q_i in the list (not neccessarily in the same order), and answering them "yes" if $a_i = 0$ and "no" if $a_i = 1$, M accepts, and

($\star\star$) for $i = 1 \ldots k$ $q_i a_i \in A \oplus \overline{A}$ and $\langle q_i a_i, z_i \rangle \in D$ and there are exactly $f(q_i)$ strings z_i', greater than or equal to z_i, such that $\langle q_i a_i, z_i' \rangle \in D$ }

We claim that $B_2 \in \mathbf{G}\text{-}K'$. Again, since for every x and for every y such that $M_y^A(x)$ accepts, there is exactly one string v such that $\langle x, y, v \rangle \in B_2$,

$$x \in L \Longleftrightarrow \mathbf{C}_{2^{p(|x|)-1}+1}\ w\ :\ \langle x, w \rangle \in B_2$$

and it is left to show that $B_2 \in \mathbf{G}\text{-}K'$. ($\star$) is a predicate that can be checked in polynomial time and condition ($\star\star$) can be written in the following way

$$\text{for } i = 1 \ldots m\ [\langle q_i a_i, z_i \rangle \in D \text{ and } \mathbf{G}_{f(q_i)}\ z_i'\ :\ (z_i \leq z_i' \text{ and } \langle q_i a_i, z_i \rangle \in D) \text{ and } \langle q_i a_i, z_i' \rangle \in D]$$

($\star\star$) is therefore an unbounded cartesian product of predicates in $\mathbf{G}\text{-}K'$ and by theorem 3.9, it is a predicate in $\mathbf{G}\text{-}K'$. It follows that $B_2 \in \mathbf{G}\text{-}K'$ and $\mathrm{PP}^{\mathbf{C}K'} \subseteq \mathbf{CG}\text{-}K'$. But by corollary 3.11, $\mathbf{CG}\text{-}K' \subseteq \mathbf{CC}K' = \mathbf{C}K$.

ii/ The proof is completely analogous to the one above being 1 instead of $2^{p(|x|)-1} + 1$ the threshold of the machine. □

Observe that from the above proof it can also be derived that every language recognized by nondeterministic or probabilistic oracle machines, with an oracle in CH, can be decided by a machine of the same type, quering the oracle just once.

5. Separations

In this section we try to show that the containments between the classes in the studied language hierarchy are strict. Absolute separations are very hard to accomplish, since they would immediately imply P≠PSPACE, solving a long time standing open problem. A more modest approach is to try to find relativized separations. These relativized separations are still important for different reasons: One of them is that these separations, together with the relativization in which PSPACE is used as oracle (forcing all the classes in CH to collapse together), show that there are contradictory relativizations for this classes, giving stronger evidence that the absolute separation problem is very hard. Quoting Hartmanis, "...the proof that a problem can be relativized in two contradictory ways serves today in theoretical computer science almost the same role as proving a problem NP hard in the study of algorithms. If a problem is NP hard, we are very unlikely to solve it in reasonable time sufficiently big instances of this problem. Similarly, the contradictory relativization

(of a sufficiently "rich" problem) is good indication that it can not be solved with our current mathematical techniques" [Har,87]. Another reason for the importance of the relativized separations of the classes in CH, is that they provide *absolute* separations for the corresponding classes in the logarithmic time counting hierarchy [To,88a].

We introduce a new technique based on counting the number of accepting computations of the machines over which we diagonalize. Lemma 5.4 is the main result in which our constructions are based; we try to motivate it with the example of the separation of NP from $\mathbf{G}_{=}$. Apart from this mentioned separation we also separate $\oplus$P from PP, and NP from $\oplus$P, ($\oplus$P is the class "parity" of languages recognized by nondeterministic polynomial time machines with an even number of computation paths for words in the language, and an odd number of accepting paths for words that are not in the language). Using the known inclusions between these classes we are able to prove other results, separating the lower levels of CH. As a consequence we obtain absolute separations for the lower levels of the logarithmic time counting hierarchy.

Although there are several references in the literature of relativizations separating counting classes from classes in the polynomial time hierarchy, [An,80], [Ya,85], [Hås,86], to our knowledge these are the first relativizations separating counting classes from other counting classes. In [Be,Gi,81] it was claimed that $\oplus\mathrm{P}^A \not\subset \mathrm{PP}^A$ for a random oracle A, but the proof of this result was incorrect.

We start separating NP from $\mathbf{G}_{=}$.

Let $M_1, M_2 \ldots$ be an enumeration of all the probabilistic Turing machines, and $p_1, p_2 \ldots$ an enumeration of the polynomials. W.l.o.g. we can suppose that for every k, M_k has computation time bounded by p_k.

Theorem 5.1: There is an oracle A such that $\mathrm{NP}^A \not\subset \mathbf{G}_{=}^A$.

Proof: For every set A, define $L_A = \{0^n : \exists w(|w| = n \text{ and } w \in A)\}$ clearly, for every set $A, L_A \in \mathrm{NP}^A$. We construct in stages a set A such that $L_A \notin \mathbf{G}_{=}^A$.

Stage 0. $A_0 := \emptyset; n_0 := 0$.

Stage s. Let n_s be the smallest integer such that

$n_s > n_{s-1}$

$n_s > \max\{p_i(n_{s-1}) : i < s\}$

$2^{n_s} > p_s(n_s)$

$(\star)$ $A_s := A_{s-1} \cup B$, being $B \subseteq \Sigma^{n_s}$, such that $B \neq \emptyset \iff 0^{n_s} \notin L(M_s, A_{s-1} \cup B)$;

(Here, for a string x and an oracle B, $x \in L(M_s, B)$ means that machine M_s has exactly th accepting computation paths for this input, being th the threshold of the machine for input x.)

Let $A = \bigcup_s A_s$. It is clear, following the same ideas as in [Ba,Gi,So,75], that if we prove the existence of set B in $(\star)$, then the set L_A is not in $\mathbf{G}^A$. In the following, we show that the set B in $(\star)$ always exists.

Notation: $Q^B_{a_1,\ldots,a_k,(b_1,\ldots,b_l)}$ denotes the number of accepting paths from M_s with oracle $A_{s-1} \cup B$ and input 0^{n_s}, in which all the words $a_1 \ldots a_k$ are queried, and none of the words $b_1 \ldots b_l$ is queried to the oracle. For example, $Q^{w_1,w_3}_{w_1,w_2,(w_3,w_4)}$ denotes the number of accepting paths from M_s with oracle $A_{s-1} \cup \{w_1, w_3\}$ and input 0^{n_s}, in which the words w_1 and w_2 are queried, and none of the words w_3, w_4 are queried to the oracle. Notice that we have omitted the "{'s" from the set notation, from the superscript of the Q. Also observe that if a word is not queried, it doesn't make any difference if we drop it into, or take it away from the oracle, for example, the above expression $Q^{w_1,w_3}_{w_1,w_2,(w_3,w_4)}$ is equivalent to $Q^{w_1}_{w_1,w_2,(w_3,w_4)}$

The following lemma is needed for proving the result. We omit the proof since it is straightforward. It just says that we can decompose the set of accepting computations quering $w_1 \ldots w_k$ into two: those that query also w_{k+1} and those that do not.

Lemma 5.2: For any set B and any $k+1$ words $w_1, \ldots w_{k+1} \in \Sigma^{n_s}$ the following equality holds:

$$Q^B_{w_1,\ldots w_k} = Q^B_{w_1,\ldots w_k,w_{k+1}} + Q^B_{w_1,\ldots w_k,(w_{k+1})}$$

(Sometimes we will use the above equality in the form $Q^B_{w_1,\ldots w_k,w_{k+1}} = Q^B_{w_1,\ldots w_k,(w_{k+1})} - Q^B_{w_1,\ldots w_k}$ which should not create any confussion).

In order to operate in a concise form, the defined Q-expressions representing the number of accepting paths with different oracles, will be grouped into a combinatorial formula. The motivation for this is presented with more detail in [To,88a].

Notation: For any $B, D \subseteq \Sigma^{n_s}$, with $B \cap D = \emptyset$,

$$J^B_D = \sum_{i=0}^{\|D\|} (-1)^i \sum_{\substack{A \subseteq D \\ \|A\|=i}} Q^{B \cup A}_D$$

This formalism is needed for our proofs since otherwise using only the "Q's", we would have to carry very long symbol strings.

We introduce now the main lemma in this section, that would enable us to prove the existence of the oracles separating the classes.

Lemma 5.3: For any sequence of words $w_1, \ldots, w_k, w_{k+1}$ in Σ^{n_s}, and any set $B \subseteq \Sigma^{n_s}$, with $B \cap \{w_1, \ldots, w_k, w_{k+1}\} = \emptyset$

$$J^{B \cup \{w_{k+1}\}}_{w_1 \ldots w_k} = J^B_{w_1 \ldots w_k} - J^B_{w_1 \ldots w_k, w_{k+1}}$$

Proof: For the proof, first we decompose the J's into Q's following the definition, and then manipulate the Q's either decomposing them into two by lemma 5.2, or deleting from the oracle some words that are not queried.

Let $D = \{w_1 \dots w_k\}$.

$$
\begin{aligned}
J_D^{B\cup\{w_{k+1}\}} &= \sum_{i=0}^{k}(-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} Q_{w_1\dots w_k}^{B\cup\{w_{k+1}\}\cup A} \\
&= \sum_{i=0}^{k}(-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} (Q_{w_1\dots w_k,w_{k+1}}^{B\cup\{w_{k+1}\}\cup A} + Q_{w_1\dots w_k(w_{k+1})}^{B\cup\{w_{k+1}\}\cup A}) \\
&= \sum_{i=0}^{k}(-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} (Q_{w_1\dots w_k,w_{k+1}}^{B\cup\{w_{k+1}\}\cup A} + Q_{w_1\dots w_k(w_{k+1})}^{B\cup A}) \\
&= \sum_{i=0}^{k}(-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} (Q_{w_1\dots w_{k+1}}^{B\cup\{w_{k+1}\}\cup A} + Q_{w_1\dots w_k}^{B\cup A} - Q_{w_1\dots w_{k+1}}^{B\cup A}) \\
&= \sum_{i=0}^{k}((-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} Q_{w_1\dots w_k}^{B\cup A}) + \sum_{i=0}^{k}(-1)^i \sum_{\substack{A\subseteq D\\ \|A\|=i}} (Q_{w_1\dots w_{k+1}}^{B\cup\{w_{k+1}\}\cup A} - Q_{w_1\dots w_{k+1}}^{B\cup A}) \\
&= J_D^B + \sum_{i=0}^{k}(-1)^i(\sum_{\substack{A\subseteq D\cup\{w_{k+1}\}\\ \|A\|=i+1}} Q_{w_1\dots w_{k+1}}^{B\cup A} - \sum_{\substack{A\subseteq D\\ \|A\|=i+1}} Q_{w_1\dots w_{k+1}}^{B\cup A} - \sum_{\substack{A\subseteq D\\ \|A\|=i}} Q_{w_1\dots w_{k+1}}^{B\cup A}) \\
&= J_D^B + \sum_{i=0}^{k}((-1)^i(\sum_{\substack{A\subseteq D\cup\{w_{k+1}\}\\ \|A\|=i+1}} Q_{w_1\dots w_{k+1}}^{B\cup A}) - Q_{w_1\dots w_{k+1}}^{B} \qquad (3) \\
&= J_D^B - \sum_{i=1}^{k+1}((-1)^i(\sum_{\substack{A\subseteq D\cup\{w_{k+1}\}\\ \|A\|=i}} Q_{w_1\dots w_{k+1}}^{B\cup A}) - Q_{w_1\dots w_{k+1}}^{B} \\
&= J_D^B - J_{D\cup\{w_{k+1}\}}^B
\end{aligned}
$$

Maybe the step taken to obtain the expression in (3) needs some clarification. Observe that in the expression before (3), the two last sums only differ in the size of $\| A \|$. Since these sums are part of another sum and are multiplied by $(-1)^i$, the terms cancel, the sum "telescopes", remaining only $Q_{w_1\dots w_{k+1}}^B$ and

$$(-1)^k \sum_{\substack{A\subseteq D\\ \|A\|=k+1}} Q_{w_1\dots w_{k+1}}^{B\cup A}$$

but this last term is 0 since $||D|| = k$. □

Lemma 5.4: For $s \geq 1$, if for every set $R \subseteq \Sigma^{n_s}$, it is true that

$$0^{n_s} \in L(M_s, A_{s-1} \cup R) \Longleftrightarrow R \neq \emptyset$$

then for any nonempty sequence of words $w_1 \ldots w_k$ in Σ^{n_s}, and any oracle B, with $B \neq \emptyset$, $B \neq \Sigma^{n_s}$ and $\{w_1 \ldots w_k\} \cap B = \emptyset$, it holds that $J^B_{w_1 \ldots w_k} = 0$. Moreover, $J^{\emptyset}_{w_1 \ldots w_k} = J^{\emptyset}_{w_1} = Q^{\emptyset} - th$

Proof: By hypothesis, and using the definition of J, for every set $B \neq \emptyset$, $J^B = Q^B = th$.

By lemma 5.3

$$J^B_{w_1 \ldots w_{k+1}} = J^B_{w_1 \ldots w_k} - J^{B \cup \{w_{k+1}\}}_{w_1 \ldots w_k}$$

We prove the first claim by induction on k.

For $k = 1$, $J^B_{w_1} = J^B - J^{B \cup \{w_1\}} = th - th = 0$

For $k > 1$, $J^B_{w_1 \ldots w_k} = J^B_{w_1 \ldots w_{k-1}} - J^{B \cup \{w_k\}}_{w_1 \ldots w_{k-1}}$ where by induction hypothesis both terms are 0.

The second claim is proved also by induction on k.

For k=1,

$$J^{\emptyset}_{w_1} = Q^{\emptyset}_{w_1} - Q^{w_1}_{w_1} = Q^{\emptyset} - Q^{\emptyset}_{(w_1)} - Q^{w_1} + Q^{w_1}_{(w_1)} = Q^{\emptyset} - Q^{w_1} = Q^{\emptyset} - th$$

For $k > 1$, $J^{\emptyset}_{w_1 \ldots w_k} = J^{\emptyset}_{w_1 \ldots w_{k-1}} - J^{w_k}_{w_1 \ldots w_{k-1}}$, but by the first part of the result, $J^{w_k}_{w_1 \ldots w_{k-1}} = 0$. By induction hypothesis $J^{\emptyset}_{w_1 \ldots w_{k-1}} = J^{\emptyset}_{w_1}$. Thus $J^{\emptyset}_{w_1 \ldots w_k} = J^{\emptyset}_{w_1}$. □

Now we are ready to prove the existence of the set B in $(\star)$.

Lemma 5.5: For every $s \geq 1$ there is a set $B \subseteq \Sigma^{n_s}$, such that $B \neq \emptyset \Longleftrightarrow 0^{n_s} \notin L(M_s, A_{s-1} \cup B)$.

Proof: Let th be the threshold of the machine for input 0^{n_s}. Suppose that the mentioned set B does not exist, then for every set $B \subseteq \Sigma^{n_s}, B \neq \emptyset, Q^B = th$. We are in the hypothesis of lemma 5.4.

Let $p = p_{n_s}(n_s)$. Since the running time of M_s on input 0^{n_s} is bounded by p, the machine can make at the most p queries to the oracle on every computation path, and therefore $J^{\emptyset}_{w_1 \ldots w_{p+1}} = 0$. (Recall that $J^{\emptyset}_{w_1 \ldots w_{p+1}}$ is a sum of computation paths in which all words $w_1 \ldots w_{p+1}$ are queried).

On the other hand, by lemma 5.4, $J^{\emptyset}_{w_1 \ldots w_{p+1}} = J^{\emptyset}_{w_1} = Q^{\emptyset} - th$. It follows that $Q^{\emptyset} = th$, which contradicts the hypothesis since $0^{n_s} \notin L(M_s, A_{s-1})$. □

Corollary 5.6: There is an oracle A such that $\mathbf{G}^A \neq \mathbf{C}^A$

Proof: Straightforward from the above separation, considering that the proof of NP$\subseteq \mathbf{C}$ relativizes. □

Corollary 5.7: There is an oracle A such that $\mathbf{G}^A$ is not closed under complements.

Proof: Follows from theorem 4.1 and the fact that co-NP is included in $\mathbf{G}_=$, and the proof relativizes. □

We present now separations dealing with the class $\oplus$P (parity). This class was defined in [Pa,Za,83]. Recently some results have appeared separating this class from the polynomial time hierarchy [Fu,Sa,Si,84], [Ya,85], [Hås,86]. We show relativizations separating PP and $\oplus$P. As a consequence, these results will imply separations in the lower levels of PH.

Definition 5.8: $\oplus$P=$\{L \subseteq \Sigma^\star$: there is a nondeterministic polynomial time machine recognizing L with an even number of accepting computation paths for input strings in L, and an odd number of accepting computation paths for input strings in $\overline{L}\}$.

We present now an oracle separating NP from $\oplus$P.

Theorem 5.9: There is an oracle A such that $\mathrm{NP}^A \not\subset \oplus\mathrm{P}^A$.

Let $M_1, M_2 \ldots$ be an enumeration of all the nondeterministic Turing machines, and $p_1, p_2 \ldots$ an enumeration of the polynomials. W.l.o.g. we can suppose that for every k, M_k has computation time bounded by p_k.

For every set A, define $L_A = \{0^n : \exists w |w| = n \text{ and } w \in A\}$. Clearly, for every set $A, L_A \in \mathrm{NP}^A$. We construct in stages a set A such that $L_A \notin \oplus\mathrm{P}^A$.

Stage 0. $A_0 := \emptyset; n_0 := 0$.

Stage s. Let n_s be the smallest integer such that
$n_s > n_{s-1}$
$n_s > \max\{p_i(n_{s-1}) : i < s\}$
$2^{n_s} > p_s(n_s)$

$(\star\star)$ $A_s := A_{s-1} \cup B$, being $B \subseteq \Sigma^{n_s}$, such that $B \neq \emptyset \iff 0^{n_s} \notin L(M_s, A_{s-1} \cup B)$;

Let $A = \bigcup_s A_s$. Observe that since we are trying to diagonalize away from parity, the expression $0^{n_s} \notin L(M_s, A_{s-1} \cup B)$ means that machine M_s on input 0^{n_s}, and oracle $A_{s-1} \cup B$ has an odd number of computation paths. It should be clear that if we manage to prove the existence of set B in $(\star\star)$, then the set L_A cannot be in $\oplus\mathrm{P}^A$. In the following, we show that the set B in $(\star\star)$ always exists, we will make use of lemma 5.3.

Lemma 5.10: For $s \geq 1$, if for every set $R \subseteq \Sigma^{n_s}$, it is true that
$$0^{n_s} \in L(M_s, A_{s-1} \cup R) \iff R \neq \emptyset$$
then for any nonempty sequence of words $w_1 \ldots w_k$ in Σ^{n_s}, and any oracle B, with $B \neq \emptyset$, $B \neq \Sigma^{n_s}$ and $\{w_1 \ldots w_k\} \cap B = \emptyset$, it holds that $J^B_{w_1 \ldots w_k}$ is even. Moreover, $J^\emptyset_{w_1 \ldots w_k}$ is odd.

Proof: By the definition of acceptance of M_s and the hypothesis, for every set B, $B \neq \emptyset$,

J^B is even, and by lemma 5.3

$$J^B_{w_1 \ldots w_{k+1}} = J^B_{w_1 \ldots w_k} - J^{B \cup \{w_{k+1}\}}_{w_1 \ldots w_k}$$

We prove the first claim by induction on k.

For $k = 1$, $J^B_{w_1} = J^B - J^{B \cup \{w_1\}}$, since J^B and $J^{B \cup \{w_1\}}$ are even, so is $J^B_{w_1}$.

For $k > 1$, $J^B_{w_1 \ldots w_k} = J^B_{w_1 \ldots w_{k-1}} - J^{B \cup \{w_k\}}_{w_1 \ldots w_{k-1}}$, and by induction hypothesis, both members of the right hand side of the equation are even.

The second claim is proved also by induction on k.

For $k = 1$, $J^\emptyset_{w_1} = J^\emptyset - J^{w_1}$. $J^\emptyset$ is odd by hypothesis and J^{w_1} is even by the first part of the result.

For $k > 1$, $J^\emptyset_{w_1 \ldots w_k} = J^\emptyset_{w_1 \ldots w_{k-1}} - J^{w_k}_{w_1 \ldots w_{k-1}}$, but by the first part of the result, $J^{w_k}_{w_1 \ldots w_{k-1}}$ is even. By induction hypothesis $J^\emptyset_{w_1 \ldots w_{k-1}}$ is odd, and it follows that $J^\emptyset_{w_1 \ldots w_k}$ is also odd. □

Now we are ready to prove the existence of set B in $(\star\star)$.

Lemma 5.11: For every $s \geq 1$ there is a set $B \subseteq \Sigma^{n_s}$, such that $B \neq \emptyset \iff 0^{n_s} \notin L(M_s, A_{s-1} \cup B)$.

Proof: Suppose that the mentioned set B does not exist, then we are in the hypothesis of lemma 5.10.

Let $p = p_{n_s}(n_s)$. Since the running time of M_s on input 0^{n_s} is bounded by p, the machine can make at most p queries to the oracle on each computation path, and therefore $J^\emptyset_{w_1 \ldots w_{p+1}} = 0$.

On the other hand, by lemma 5.10, $J^\emptyset_{w_1 \ldots w_{p+1}}$ is odd. This is a contradiction and it follows that the mentioned set B always exists. □

Corollary 5.12: There is an oracle A such that $\mathrm{PP}^A \not\subset \oplus\mathrm{P}^A$.

Proof: Straightforward considering that the proof of NP⊆PP relativizes. □

We present now the last separation, this time separating $\oplus$P from PP. This result will bring as a consequence the separation of different classes in the counting hierarchy.

Theorem 5.13: There is an oracle A such that $\oplus\mathrm{P}^A \not\subset \mathrm{PP}^A$.

Let $M_1, M_2 \ldots$ be an enumeration of all the probabilistic Turing machines, and $p_1, p_2 \ldots$ an enumeration of the polynomials. W.l.o.g. we can suppose that for every k, M_k has computation time bounded by p_k.

For every set A, define $L_A = \{0^n : ||A \cap \Sigma^n|| \text{ is even}\}$. Clearly, for every set $A, L_A \in \oplus\mathrm{P}^A$. We construct in stages a set A such that $L_A \notin \mathbf{C}^A$.

Stage 0. $A_0 := \emptyset; n_0 := 0$.

Stage s. Let n_s be the smallest integer such that
$n_s > n_{s-1}$
$n_s > \max\{p_i(n_{s-1}) : i < s\}$
$2^{n_s} > p_s(n_s)$

$(\star\star\star)$ $A_s := A_{s-1} \cup B$, being $B \subseteq \Sigma^{n_s}$, such that $0^{n_s} \in L(M_s, A_{s-1} \cup B) \iff ||B||$ is odd;

Let $A = \bigcup_s A_s$. In the following, we show that the set B in $(\star\star\star)$ always exists, which implies that $L_A \notin \mathrm{PP}^A$.

Lemma 5.14: For $s \geq 1$, if for every set $R \subseteq \Sigma^{n_s}$, it is true that
$$0^{n_s} \in L(M_s, A_{s-1} \cup R) \iff ||R|| \text{ is even}$$
then for any nonempty sequence of words $w_1 \dots w_k$ in Σ^{n_s}, and any oracle B, with $B \neq \Sigma^{n_s}$ and $\{w_1 \dots w_k\} \cap B = \emptyset$, it holds that $J^B_{w_1 \dots w_k} \neq 0$. More precisely, if $||B||$ is even then $J^B_{w_1 \dots w_k} > 0$ and if $||B||$ is odd then $J^B_{w_1 \dots w_k} < 0$.

Proof: By the definition of acceptance of M_s and the hypothesis, for every set B, $||B||$ is even $\iff J^B \geq th$, and by lemma 5.3

$$J^B_{w_1 \dots w_{k+1}} = J^B_{w_1 \dots w_k} - J^{B \cup \{w_{k+1}\}}_{w_1 \dots w_k}$$

By induction on k. (Suppose $||B||$ is even, the odd case is analogous.)
For $k = 1$, $J^B_{w_1} = J^B - J^{B \cup \{w_1\}}$, since $J^B \geq th$ and $J^{B \cup \{w_1\}} < th$, we obtain $J^B_{w_1} > 0$.
For $k > 1$, $J^B_{w_1 \dots w_k} = J^B_{w_1 \dots w_{k-1}} - J^{B \cup \{w_k\}}_{w_1 \dots w_{k-1}}$, and by induction hypothesis $J^B_{w_1 \dots w_{k-1}} > 0$ and $J^{B \cup \{w_k\}}_{w_1 \dots w_{k-1}} < 0$. It follows that $J^B_{w_1 \dots w_k} > 0$. □

Now we are ready to prove the existence of set B in $(\star\star\star)$.

Lemma 5.15: For every $s > 1$ there is a set $B \subseteq \Sigma^{n_s}$, such that
$$0^{n_s} \in L(M_s, A_{s-1} \cup B) \iff ||B|| \text{ is odd}$$

Proof: Let th be the threshold of the machine for input 0^{n_s}. Suppose that the mentioned set B does not exist, then we are in the hypothesis of lemma 5.14.

Let $p = p_{n_s}(n_s)$. Since the running time of M_s on input 0^{n_s} is bounded by p, the machine can make at most p queries to the oracle on each computation path, and therefore $J^{\emptyset}_{w_1 \dots w_{p+1}} = 0$.

On the other hand, by lemma 5.14, $J^{\emptyset}_{w_1 \dots w_{p+1}} > 0$. This is a contradiction and it follows that the mentioned set B always exists. □

Corollary 5.16: There is an oracle A such that $\exists \mathbf{C}^A \neq \mathbf{C}^A$ and $\forall \mathbf{C}^A \neq \mathbf{C}^A$.

In [To,88a] a counting hierarchy of classes operating in logarithmic time has been defined. It is not hard to prove that the above relativized separations imply absolute separations for the lower levels of the logarithmic time counting hierarchy.

6. Conclusions and further research areas

Our work has been motivated by the study of a hierarchy connected with the idea of counting: the polynomial time counting hierarchy. We have studied the closure of the classes in CH under boolean operations and unbounded cartesian product, showing that for these properties, this hierarchy behaves in a different way as PH. Using these results we have given an oracle characterization of CH, parallel to the one existing for the polynomial time hierarchy, closing an open problem and unifying concepts. From the oracle characterization of CH, follows also that probabilistic oracle machines can be simmulated by machines of the same type querying a new oracle once at the most on every computation path.

Finally in section 5, we have separated some of the studied classes. Three relativizations have been given, separating NP from $\mathbf{G}_{=}$, NP from $\oplus$P, and $\oplus$P from PP. These relativizations provoke other separations, such as $\mathbf{G}_{=}$ from $\mathbf{C}$, $\exists\mathbf{C}$ and $\forall\mathbf{C}$ from $\mathbf{C}$, etc., as well as some other classes related with the closure under boolean operations of the counting classes. The relativized separations for the classes in CH imply absolute separations for the corresponding logarithmic time classes, and thus, we have separated the lower levels of the logarithmic time counting hierarchy.

Although we have obtained many new results, solving some open problems, there are still several questions connected with the counting hierarchies, that remain open. In the following we give a list of some of these questions. We will not include in this list obvious open problems of type $\mathrm{P} \stackrel{?}{=} \mathrm{PP}$ or $\mathrm{NP} \stackrel{?}{=} \mathrm{PP}$, which we believe are still far from being solved; we will concentrate more in problems that apparently can be solved with the existing techniques of structural complexity theory (or at least they do not seem so far as the ones mentioned in the first place).

If the class NP is closed under complements then the polynomial hierarchy collapses; does the counting hierarchy collapse if PP is closed under intersection? Observe that in theorem 3.12 we have shown a collapse of the class $\mathrm{P}^{\mathrm{PP}[(\log n)]}$ to PP in case PP is closed under unbounded cartesian product, a certain kind of unbounded intersection.

Recently, Toda [Tod,89] has obtained a remarkable result showing that $\mathrm{PH} \subseteq \mathrm{P}^{\mathrm{PP}}$. This implies that if $\mathbf{C}$ is included in PH, then the polynomial time hierarchy collapses. Is this fact also true if $\mathbf{G}_{=} \subseteq$PH?

The obvious open problem related with the last section is to know if there is an oracle separating every classes in the counting hierarchy, in analogy with the result for PH [Ya,85], [Hås,86], [Ko,87]. This question is closely related to the existence of exponential lower bounds for constant depth circuits made of threshold gates, and seems to need new techniques. Nevertheless there are interesting relativization questions that still remain

open and might be easier; for example, is there an oracle for which PP is not closed under intersection?

References

[An,80] D. Angluin: On counting problems and the polynomial-time hierarchy. *Theoretical Computer Science* 12 (1980), 161–173.

[Ba,Gi,So,75] T.P. Baker, J. Gill, and R.M. Solovay: Relativizations of the P=?NP question. *SIAM Journal of Computing* 4 (1975), 431–442.

[Ba,Di,Ga,88] J.L. Balcázar, J. Diaz, and J. Gabarró: *Structural Complexity* (vol. I). Springer-Verlag (1987).

[Be,Gi,82] C.H. Bennet and J. Gill: Relative to a random oracle A $\mathrm{p}^A \neq \mathrm{NP}^A \neq \text{co-NP}^A$ with probability 1. *SIAM Journal of Computing* 10 (1981) 96–112.

[Ch,Ko,St,81] A.K. Chandra, D.C. Kozen, L.J. Stockmeyer: Alternation. *Journal of the ACM* 28 (1981), 114–133.

[Fu,Sa,Si,84] M. Furst, J.B. Saxe, M. Sipser: Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory* 17 (1984), 13–27.

[Gi,77] J. Gill: Computational complexity of probabilistic Turing machines. *SIAM Journal of Computing* 6 (1977), 675–695.

[Har,87] J. Hartmanis: The structural complexity column: Sparse complete sets for NP and the optimal collapse of the polynomial hierarchy. *Bull. EATCS* 32 (1987), 73–81.

[Hås,86] J. Håstad: Computational limitations for small depth circuits. Ph.D. Thesis, M.I.T. (1986).

[Ko,87] K. Ko: Relativized polynomial time hierarchies having exactly K levels. Proc. 3rd Structure in Complexity Theory Conference (1988) 251–252.

[Pa,83] C.H. Papadimitriou: Games against nature. Proc. 24th FOCS (1983), 446–450.

[Pa,Za,83] C.H. Papadimitriou, S. Zachos: Two remarks on the power of counting. 6th GI Conference on Theoret. Comput. Sci., Lect. Notes in Comp. Sci. 145 (1983), Springer-Verlag, 269–276.

[Ru,85] D.A. Russo: Structural properties of complexity classes. Ph.D. Thesis, University of California, Santa Barbara (1985).

[Sc,85] U. Schöning: *Complexity and structure.* Lect. Notes in Comp. Sci. 211, Springer-Verlag (1985).

[Sim,75] J. Simon: On some central problems in computational complexity. Ph.D. Thesis, Cornell University (1975).

[St,77] L.J. Stockmeyer: The polynomial time hierarchy. *Theoretical Computer Science* 3 (1977), 1–22.

[Tod,89] S. Toda: On the computational power of PP and $\oplus$P, Proc. 30th FOCS, (1989) 514–519.

[To,88a] J. Torán: Structural properties of the counting hierarchies. Ph.D. Thesis. Facultat d'Informática de Barcelona, (1988).

[To,88b] J. Torán: An oracle characterization of the counting hierarchy. Proc. 3rd Structure in Complexity Theory Conference (1988) 213–223.

[Va,79] L.G. Valiant: The complexity of computing the permanent. *Theoretical Computer Science* 8 (1979), 189–201.

[Wa,86] K. Wagner: The complexity of combinatorial problems with succinct input representation. *Acta Informatica* 23 (1986), 325–356.

[Wa,We,86] K. Wagner and G. Wechsung: *Computational complexity*. Reidel (1986).

[Wr,77] C. Wrathall: Complete sets and the polynomial time hierarchy, *Theoretical Computer Science*3 (1977) 23–33.

[Ya,85] A.Yao: Separating the polynomial time hierarchy with oracles, Proc. 26th FOCS, (1985), 1–10.